(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 August 2001 (02.08.2001)

PCT

(10) International Publication Number
WO 01/55917 A1

(51) International Patent Classification[7]: G06F 17/50

(21) International Application Number: PCT/US01/02982

(22) International Filing Date: 29 January 2001 (29.01.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/492,634 27 January 2000 (27.01.2000) US

(71) Applicant: MORPHICS TECHNOLOGY INC. [US/US]; Suite 100, 675 Campbell Technology Parkway, Campbell, CA 95008 (US).

(72) Inventors: SUBRAMANIAN, Ravi; 150 Alley Way, Mountain View, CA 94040 (US). RIEKEN, Keith; 21603 La Playa Court, Cupertino, CA 95014 (US).

(74) Agents: MORRIS, Francis, E. et al.; Pennie & Edmonds LLP, 1155 Avenue of the Americas, New York, NY 10036 (US).

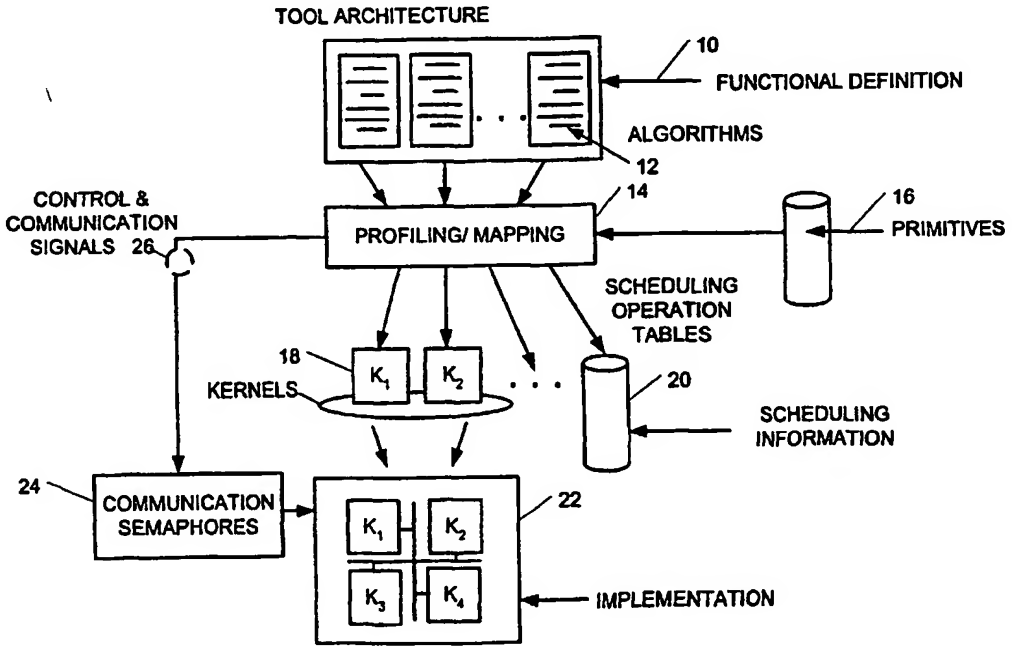
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) Title: IMPROVED APPARATUS AND METHOD FOR MULTI-THREADED SIGNAL PROCESSING



(57) Abstract: System and circuit design methodology and apparatus implements general functional definition (10) using multi-threaded representation thereof, which may be profiled for parallel processing using one or more corresponding kernel logic elements (18). Preferably, communication (26), networking, or media processing functionality or algorithm (12) is functionally analyzed and symbolically represented to identify one or more thread segments, which are each profiled (14) using temporal and/or non-temporal functions, according to one or more particular fixed, parameterizable, programmable, or reconfigurable logic kernel.

WO 01/55917 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

IMPROVED APPARATUS AND METHOD FOR MULTI-THREADED SIGNAL PROCESSING

Field of Invention

Invention relates to electronic data and signal processing, particularly to high-performance multi-threaded information processing techniques.

Background of Invention

Traditional methods for achieving high-performance in computational systems for digital information processing have centered around the design of architectures that deliver greater levels of parallelism. This is typically achieved via the design of processors and instruction-set architectures that allow for the exploitation of hardware parallelism and software concurrency.

High-performance is typically defined as the ability to execute a very large number of operations per second. This figure of merit is strongly dependent on the type of operations, which typically depends on the type of application targeted.

Traditional design of high-performance information processing systems usually relies on principles of computer architecture to define several key attributes of the processing system:

- *Instruction-set architecture* refers to the actual programmer-visible sets of instructions, and serves as the boundary between hardware and software.
- *Organization* refers to high-level aspects of computer design, such as memory system, bus structure, and internal CPU design.
- *Hardware* refers to specific detailed logic design, circuit implementation, and packaging.

In order to achieve high-performance, which is an attribute typically required in special-purpose processors (i.e., built for special applications), three approaches are taken:

(1) Instruction-level parallelism: this approach, which exploits parallelism in hardware, provides for parallel threads of processing via the use of a very long or vectorized instruction word, whose fields can be decomposed into concurrent processing threads. The mechanism to exploit this parallelism may be realized via a scheduler, which schedules operations onto one of several datapath processing units. This scheme has many drawbacks, including the difficulty of building the scheduler and identifying enough parallelism to achieve desired throughput.

(2) Superscalar techniques: this approach exploits fine-grain highly-pipelined, single-threaded processor architectures to achieve high performance. This scheme may achieve very high performance, but only for a small class of operations. For operations not well-matched to a particular datapath architecture, performance of superscalar design is reduced significantly. Thus, the superscalar approach is unsuitable for wide-ranging applications with high signal-processing content.

(3) Memory hierarchy techniques: to hide latency of memory accesses to slower memories, memory hierarchy techniques have been used extensively, especially in microprocessor designs, to increase overall system performance by intelligently using fast memories, i.e., caches,

between the processor units and slower memory effectively to hide latency of slower memory.

Conventionally, multi-processor systems may employ multi-threaded processing to improve compute performance. Multi-threading generally is a known approach for enhancing compute resource utility, and thus, overall processing performance. However, ordinary multi-threaded processing solutions are implemented using complex distributed or networked computer nodes, which are often not easily reconfigurable at lower logic or circuit level, nor contemplated for addressing advanced functional problem sets, such as multi-mode telecommunications algorithms or networking protocols. Accordingly, there is a need for improved multi-thread processing solution.

Summary of Invention

Invention resides in design and implementation methodology, processor architecture, and system for processing multi-threaded digital information (signal or data representation) to improve functional performance. Preferably, general system design or functional definition, algorithm, electronic signal, or data file is provided initially to include one or more multi-threaded representation. Such initial prototype design or function may then be profiled or otherwise characterized for parallel or effectively similar processing, in particular, in order functionally to use or otherwise be implemented in one or more corresponding fixed, parameterizable, programmable, or configurable logic units or other equivalent functional signal-processing kernel or element, using temporal and/or non-temporal functional considerations.

Preferably, relatively complex system functionality, such as for application to digital communications and/or networking and/or media processing system design, is analyzed according to pre-specified system design rules, mathematical operations, sequences of operations, or parameters, and then symbolically or schematically represented to identify one or more algorithms, specific sequences of operations, patterns of memory accesses, or segments (i.e., single or multi-"threads"), which may each be profiled, structured, or otherwise characterized for optimized operation or implementation using one or more particular fixed, parameterizable, programmable, or configurable logic unit or kernel elements. Such element is built by providing a datapath, whose structure and configurability is determined via profiling, a sequencer/finite-state-machine, whose structure and configurability is determined via profiling, and local memory, whose structure is determined via profiling memory accesses and using locality to derive local memory properties. Optionally, one or more kernel elements are implemented entirely in software or programmable logic, or combination thereof. Further, as described herein, term "profiling" refers generally to automated and/or manual processing of one or more system or function modules to define one or more configurable structures associated with each module.

Brief Description of Drawings

FIG. 1 is a general methodology and tool architecture diagram for implementing in software and/or hardware a preferred embodiment of the present invention.

FIGs. 2A-B are functional block diagrams for implementing one aspect of the present invention.

FIG. 3 is a representative functional diagram illustrating heterogeneous aspect of the present invention.

FIG. 4 is a representative functional diagram illustrating reconfigurable aspect of the present invention.

FIG. 5 is a representative functional diagram illustrating kernel aspect of the present invention.

FIG. 6 is a representative functional diagram illustrating interface aspect of the present invention.

FIG. 7 is a system methodology flow chart showing functional operations for implementing one or more aspects of the present invention.

FIG. 8 is representative of software code stubs for implementing one or more aspects of the present invention.

FIG. 9A-B are representative functional diagrams of one or more applications of present invention.

Detailed Description of Preferred Embodiment

Present innovation enables automated design and implementation to process single or multi-threaded or equivalently partitioned processing of digital data, signals, or functional representation for improved processing performance. Initially, system design or functional definition, algorithm, electronic signal, or data file provides certain single or multi-threaded representation, whereupon one or more system design or function modules are profiled, structured, or otherwise characterized for parallel or concurrent processing.

For example, multi-threaded prototype may be used or otherwise be implemented in fixed, parameterizable, programmable, or configurable logic unit or other signal-processing kernel or element. Hence, complex system functionality, such as digital communication, networking, or multi-media application, may be analyzed per system design rules, mathematical operations, sequences of operations, or parameters, then symbolically or schematically represented to identify certain single or multi-thread algorithms, specific sequences of operations, patterns of memory accesses, or segments, each thread being profiled or characterized to optimize operation or implementation using fixed, parameterizable, programmable, or configurable logic unit or kernel element.

Optionally, datapath structure is configured into single or multi-thread element, as determined by profiling, a sequencer and/or equivalent finite-state-machine, whose structure and configurability is determined by profiling, and local memory, whose structure is determined by profiling memory accesses and locality to derive memory properties.

As used herein, profiling terminology is understood to refer generally to any computer-automated and/or manual processing, interpretation, or classification of one or more system or function modules to define or categorize one or more configurable structures associated with each module, e.g., by selecting or assigning one or more functional elements or design objects, such as interconnection, signals, logic, circuits, etc. Preferably, profiling is accomplished according to one or more previously and/or dynamically defined criteria or functional rule set.

Generally, in a computer-automated and/or manual development approach, a single or multi-threaded design is processed by providing initially a first-level functional definition representing a prototype system, such that an other-level functional definition symbolically representing equivalent functionality may be generated or effectively profiled therefrom. In this hierarchical design scheme, the generated symbolic representation may identify certain threads associated with the system design, preferably at one or more functional levels.

Each thread may be profiled for processing by corresponding kernel element(s), and one or more common set of operations is identified for given threads, (e.g., on a 1-to-1, multiple-to-1, or 1-to-multiple thread-to-kernel relationship). Each thread may further be mapped to identify the sequence, or scheduling information, for each set of operators utilized to implement system or functional modules, such as a sequence of arithmetic operations, control operations, and/or memory access operations or related memory locations.

Hence, using the present system development methodology, a multi-threaded processing architecture may substantially include a set of kernel elements, such that one kernel element processes certain function represented by corresponding thread, and another kernel element in the same prototype design processes other function represented by other corresponding thread. In this partitioned or distributed processing approach, each thread may be profiled separately or hierarchically for appropriate multi-level or functional group processing. For example, a first-level or group kernel element and a second-level or group kernel element, respectively are associated with a corresponding first thread and second thread in a given function or system design.

In a representative system design for wireless code division multiple access (CDMA) communications application, it is contemplated that various kernels may be provided to serve different functional groups, such as: front-end processing (e.g., data switch selector, sample interpolation, etc.); chip-rate processing (e.g., sample epoch selection, matched filter, generic despreader, generic dechannelizer, code generation unit, integrate and dump, generic searcher control, etc.); symbol sequence processing (e.g., transport format decoder, dynamic spreading factor computer, fast Hadamard transform, etc.); channel element processing (e.g., alignment/deskewing, combiner, soft decision computer, interpath interference equalizer, receive antenna diversity combiner, etc.); interleaving (e.g., deinterleaver controller); and channel coding (e.g., turbo decoder, convolutional decoder, etc.).

Generally, present approach enables one or more functional or system designs to be implemented efficiently, preferably via current multi-threading scheme, in a single processor architecture by re-parameterizing, reprogramming, or reconfiguring kernel elements (i.e., as determined by profiling technique as described further therein,) from which corresponding threads are assembled, and/or by changing sequence of operations (i.e., as determined by mapping and/or scheduling) with which threads are implemented. Preferred embodiment implements functional or system design in one or more heterogeneous and reconfigurable logic or kernel elements (i.e., according to so-called "DRL" process, as described further herein.)

FIG. 1 is a general architecture or system block diagram showing top-level overview of present design methodology, functional modules, and software and/or hardware tool architecture, preferably implemented in one or more electronic design

automation platforms, including one or more stand-alone or networked computers, processors, engineering workstations, or other compute facility having appropriate operating system, user interface, storage management, communications interfaces, and other computer-aided design and engineering tools. Preferably, it is contemplated that present design methodology serves to provide a tool architecture and processor implementation and architecture, or data file representative thereof, for enabling system architecture, such as network implementation.

As shown, initially one or more functional definition files 10, such as design netlist, or high-level description language (such as C or HDL) defining one or more functional modules or algorithms 12 is provided manually or computed automatically. In accordance with one aspect of present implementation, functionally-selective profiling and mapping scheme 14 is processed or applied to primitives 16 and functional definitions 10 to generate or provide, particularly on a multi-threaded basis, one or more control and communication signals 26 and kernels 18. Further, profiling and mapping 14 provides scheduling data for schedule operation tables 20. Control and communication signals are processed according to one or more predefined or selected functional rule set or signaling flags, e.g., communication semaphores 24. Various kernels 18 are processed and interconnected for implementation 22, for example, in reconfigurable form as described herein for multi-threaded signal processing.

FIGs. 2A-B functional block diagrams show representative set of kernels 18, 28 and their physical implementation, including schedule and allocate function 30. Preferably, one or more kernel 18 is associated with or corresponds to profiled and

mapped thread, and is implemented reconfigurably using sequencer 32, datapath 34, and memory 36.

Hence, according to present system and circuit design methodology and/or computing apparatus, general functional definition is implementable using single or multi-threaded representation thereof, which may be profiled effectively for parallel processing using one or more corresponding kernel logic elements (e.g., according to 1-to-multi, 1-to-1, multi-to-1 or multi-to-multi kernel to thread relationship.) For example, communication, networking, or media processing functionality or algorithm is functionally analyzed and symbolically represented to identify one or more thread segments, which are each profiled or otherwise characterized for optimized operation or implementation using one or more particularly designated fixed, parameterizable, programmable, or reconfigurable logic kernel.

FIG. 3 functional diagram shows representative heterogeneous, reconfigurable, multi-processing arrangement, for example, whereupon kernel 8 may implement "small" granularity threaded function, and kernel 6 may implement "large" granularity threaded function. In this reconfigurable arrangement, various levels of functional granularity, which is preferably an attribute of design function and corresponding kernel, may be implemented or dynamically reconfigured according to design requirement or profile mapping preference.

For further illustration, FIG. 4 functional diagram shows one or more representative or available configurable logic or functions which may be employed according to present approach for implementing single or multi-threads into designated kernels, such as reconfigurable logic or programmable function units (PFU) 40 having programmable logic elements and switch matrix (e.g., for encoding

bit-level operations), reconfigurable datapaths 42 having multiplexers, registers, adders, buffers, etc. and configurable signal flow through these elements (e.g., for dedicated datapath filters), reconfigurable arithmetic 44 having address generators, memory, memory address control, etc. (e.g., for arithmetic convolution kernels), and reconfigurable control 46 having data memory, datapath, program memory, instruction decoder and controller, etc. (e.g., for real-time operating system process management).

Moreover, as further illustration of sample kernel implementation, FIG. 5 functional diagram shows preferred functional elements for implementing kernel 18, including data sequencer 32, data memory 36, and parameterizable configurable arithmetic logic unit (ALU) 34.

FIG. 6 is a representative functional diagram illustrating optional interface between dynamically reconfigurable logic (DRL) process 64 and associated configuration database for processing functions externally to main processor hardware model 50. Preferably, DRL process is heterogeneous and reconfigurable, and implemented using current innovation. As shown, hardware interfaces 54 couples processor element 52 associated with library 62 and specified functional modules 60, including processor software model 57 having C-program model 56 and input/output device drivers 58 to external DRL process 64.

In this optional embodiment, one or more single or multi-threaded digital information (e.g., signal or data representation), such as general system design or functional definition, algorithm, electronic signal or data file is provided initially to include one or more multi-threaded representation, and such initial prototype design or function is profiled or otherwise characterized for parallel or effectively similar

processing, in particular, in order functionally to use or otherwise be implemented in one or more corresponding fixed, parameterizable, programmable, or configurable logic unit or other equivalent functional signal-processing kernel or element in processor model 50, 57 for functional cooperation or emulated real-time signal interaction with external DRL process 64.

FIG. 7 flow chart shows another aspect of present operational steps. Initially, user-generated or computer-generated functions are defined 70 for prototype or other system design. Then, one or more mathematical analysis or design performance optimization scheme may be applied 72 to initial design definition. Next, one or more constituent algorithms for design definition is provided 74, and representation of such algorithms is thereby coded 76, preferably in high-level, register transfer, or behavioral functional format.

Algorithms may be profiled and mapped 78, or otherwise functionally defined or categorized manually and/or automatically for optimized or directed operation or implementation of system design modules, functions, signals, components, or other element thereof using correspondingly defined kernels 80, preferably using one or more specified design building-blocks, i.e., primitives 86. Profiling and mapping data also are provided for communications semaphores 84 and scheduling and finite state machine control and parameters 88. Then, kernel definition 80 and FSM control parameterization and scheduling 88, as well as communications semaphores 84 are applied to implement single or multi-threaded elements of present design into processor architecture with reconfigurable kernel elements 82. FIG. 8 shows representative software code of sample design indicating usage of multi-thread kernels 90.

In accordance with one aspect of present invention, profiling processing or reconfigurable algorithms representative thereof is temporal, thereby including determination of certain time value or degree of change over time. Example of temporal application includes changes in receiver algorithms required in a cellular wireless system and any associated signal processing scheme for these algorithms which can take advantage of present profiling methodology. In this example, whereupon processing throughput requirements in one path (e.g., reception direction) may increase or decrease as processing progresses (e.g., from antenna to final retrieved data representation,) present profiling scheme serves to determine hardware-software or other functional partitioning of overall design implementation.

Further, in such cellular wireless example, it is contemplated that multiple methods may perform similar or equivalent signal processing, but result in different air-interface requirements or effective functionality. Particularly in the hardware partition of a given system, various processing forms or functional elements may occur or operate at various rates. Because variable processing rates may be required, and various modes of operational control may be dictated by support for multiple processing streams, several additional non-temporal and temporal profiling techniques may be applied to provide optimal functional flexibility in view of available operational performance point or capacity of such hardware architecture (e.g., real-time and non-real-time profiling). It is contemplated generally herein that other examples of application of present innovation may arise additionally with cellular wireless, including fixed-wireless, unlicensed wireless LANs, cordless telephony, telemetry, and the like.

One profiling technique applies to hardware-based algorithms across multiple modes of operation to determine type and number of operations and storage elements required, thereby enabling designer to classify each temporally-distinct function in a form which facilitates identification of commonly-used resources.

Another profiling technique applies for controlling multiple levels of hardware definition according to frequency of change, which is required. Here, mode-dependent changes in receive path of wireless receiver, for example, may need to change at startup for global reconfiguration between transaction configuration (e.g., where transactions are multi-second transactions), and within sub-second transaction across blocks of data (e.g., "on the fly.")

Depending on profiling results, appropriate level of configurable implementation may be selected, such as for processing data at highest data rate needing control on per-cycle basis. However, flexibility may be required for control, and programmable state machine may provide optimal flexibility meeting necessary performance requirements. For a datapath which may need to be selected at configuration time, but is not changed often, then programmable interconnect may be appropriately applied.

Moreover, if datapath selection occurs real-time, then datapath-cell-based multiplexing structure may apply. Also, for control functions where operation ordering is necessary, then parameterized kernels for processing operations may apply. Additionally, in cases of high-performance requirements and low flexibility requirements, dedicated datapaths are applicable to optimize silicon implementation. In case of multi-standard wireless receiver design, which delivers optimal flexibility

relative to performance point, one or more of foregoing profiling techniques are applicable.

FIG. 9A shows general aspects of applying present invention, including flow for transferring configuration table 92 of capability, parameters and values according to one or more industry or proprietary standards through applications programming interface (API) 94 to provide one or more configuration parameters for single or multi-threaded reconfigurable system implementation according to present scheme, e.g., using wired and/or over-the-air wireless network download or other transmission/reception.

Preferred implementation receives configuration parameters through API 94 to define or implement one or more interconnected block modules 96, representing microprocessor, digital signal processor (DSP), application specific integrated circuit (ASIC), field programmable gate array (FPGA), DRL, or other functional block module, which further may be defined or implemented in one or more interconnected kernel elements 98. In accordance with one aspect of present invention, one or more configurable parameters 100 may be defined or implemented to correspond in threaded fashion to one or more specified kernel elements. Hence, in this configurable-parameter case, design and implementation method or system serves to process multi-threaded digital signal or data for improved functional performance.

Generally, system design or functional definition, algorithm, electronic signal or data file is provided to include such multi-threaded representation, and initial prototype function is thus profiled for parallel processing by one or more thread, for example, to implement certain parameterizable kernel elements, which may be constrained temporally.

More particularly, in digital wireless communication application, as shown in FIG. 9B, portable mobile radio handsets 102 transmit and receive signals wirelessly with base station 104, possibly coupled to other handsets 102 and base stations 104 through digital network 106. In this networked application, specified design rules, operations, or parameters, as well as any symbolic or schematic representation thereof identify or correspond to multi-threads, for profiling and implementation in programmable kernels or software modules.

Optionally, kernel elements may be configured for operation in base station 104 and/or handset units 102. In particular, kernels may be configured for profiled datapath, sequencer/finite-state-machine, memory, or other logical structure, possibly according to temporal or non-temporal design constraint.

Foregoing described embodiments of the invention are provided as illustrations and descriptions. They are not intended to limit the invention to precise form described.

In particular, Applicant contemplates that functional implementation of invention described herein may be implemented equivalently in hardware, software, firmware, and/or other available functional components or building blocks. Other variations and embodiments are possible in light of above teachings, and it is thus intended that the scope of invention not be limited by this Detailed Description, but rather by Claims following.

Claims

What is claimed is:

1. In a computer-assisted design system, an automated method for processing multi-threaded system functionality, the method comprising the steps of:

providing a first function definition representing a system design;

generating from the first function definition a second function definition representing symbolically the first function definition, such symbolic representation identifying one or more thread associated with the system design; and

profiling each thread for processing by a specified kernel element or set thereof.

2. The method of Claim 1 further comprising the steps of:

identifying a common sequence of operations in a given thread; and

associating the common sequence of operations with a set of operators.

3. The method of Claim 2 further comprising the step of:

associating the set of operators with a sequence of arithmetic operations.

4. The method of Claim 2 further comprising the step of:

associating the set of operators with a sequence of control operations.

5. The method of Claim 2 further comprising the step of:

associating the set of operators with a sequence of memory access operations or locations.

6. The method of Claim 1 wherein:

one or more threads is profiled according to a temporal function.

7. Apparatus for multi-threaded processing comprising:

a first kernel element; and a second kernel element;

wherein the first kernel element processes a first function represented by a first thread, the second kernel element processes a second function represented by a second thread, the first thread and the second thread each being profiled for processing respectively by the first kernel element and the second kernel element, and the first thread and the second thread being associated with a common function.

8. The apparatus of Claim 7 wherein:

a common sequence of operations is identifiable with a given thread, the common sequence of operations being associated with a set of operators.

9. The apparatus of Claim 8 wherein:

the set of operators is associated with a sequence of arithmetic, control, or memory access operations.

10. The apparatus of Claim 7 wherein:

the first or second thread is profiled according to a temporal constraint.

11. The apparatus of Claim 7 wherein:

the first and second kernel elements are implemented as one or more executable software modules.

12. The apparatus of Claim 7 wherein:

the first and second kernel elements are implemented as one or more functional modules in a fixed base station or a mobile handset of a radio communication system.

13. In a communication system comprising a base station and one or more portable units, wherein each portable unit may communicate wirelessly through radio signals with the base station, a method for signal processing comprising the step of:

generating by a base station a first signal representing a system configuration, the first signal representing symbolically one or more function definition associated with one or more thread in the system configuration, wherein each thread is profiled for processing by a specified kernel element in a portable unit.

14. The method of Claim 13 further comprising the step of:

receiving the first signal by the portable unit, one or more kernel element in the portable unit being configured to process one or more thread in the system design according to the first signal.

15. The method of Claim 13 wherein:

one or more thread is profiled according to a temporal functional constraint.

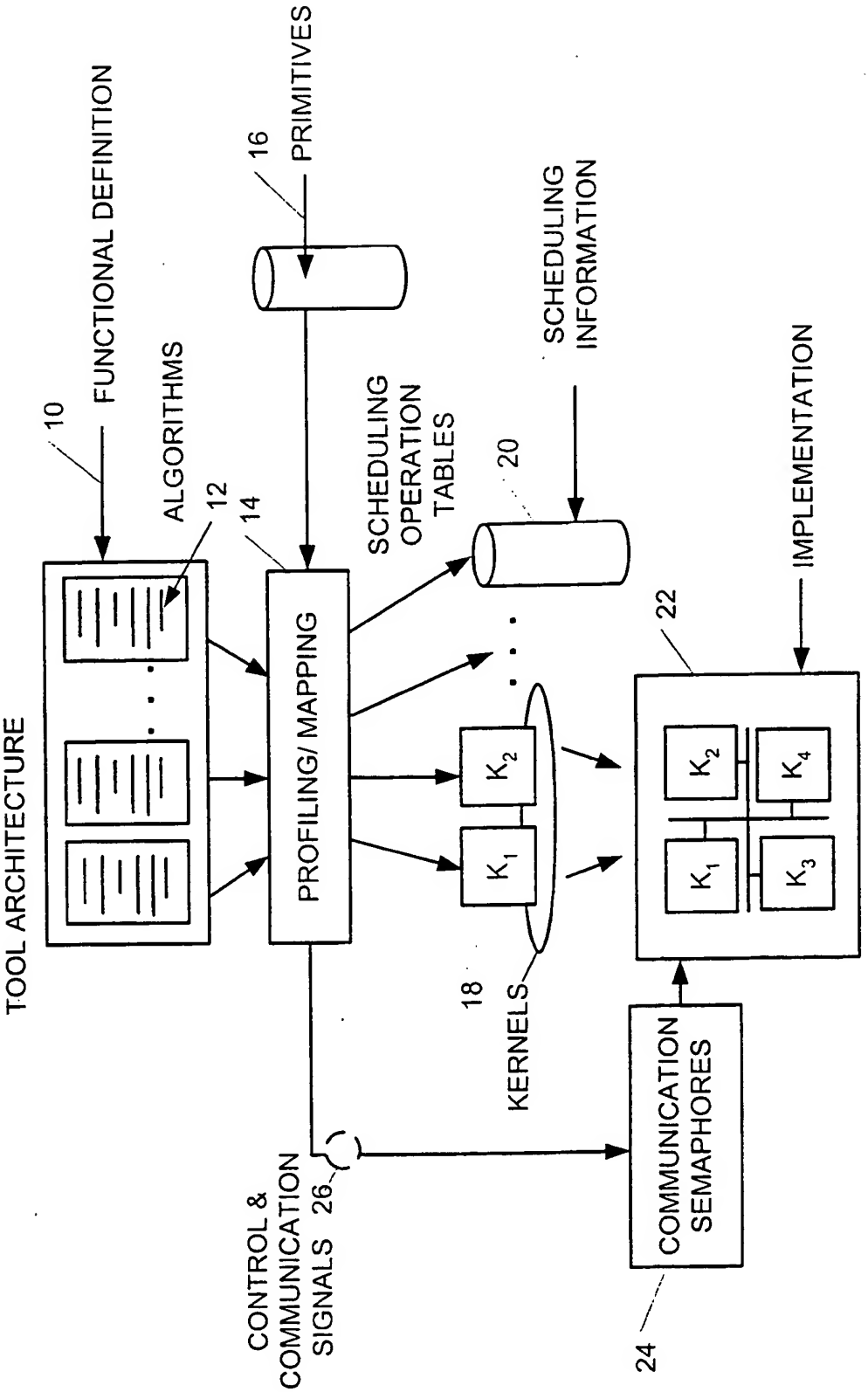

TOOL ARCHITECTURE
10
FUNCTIONAL DEFINITION
ALGORITHMS
12
14
PROFILING/ MAPPING
16
PRIMITIVES
SCHEDULING OPERATION TABLES
20
SCHEDULING INFORMATION
IMPLEMENTATION
22
K_1
K_2
K_3
K_4
18
KERNELS
CONTROL & COMMUNICATION SIGNALS 26
COMMUNICATION SEMAPHORES
24

FIG. 1

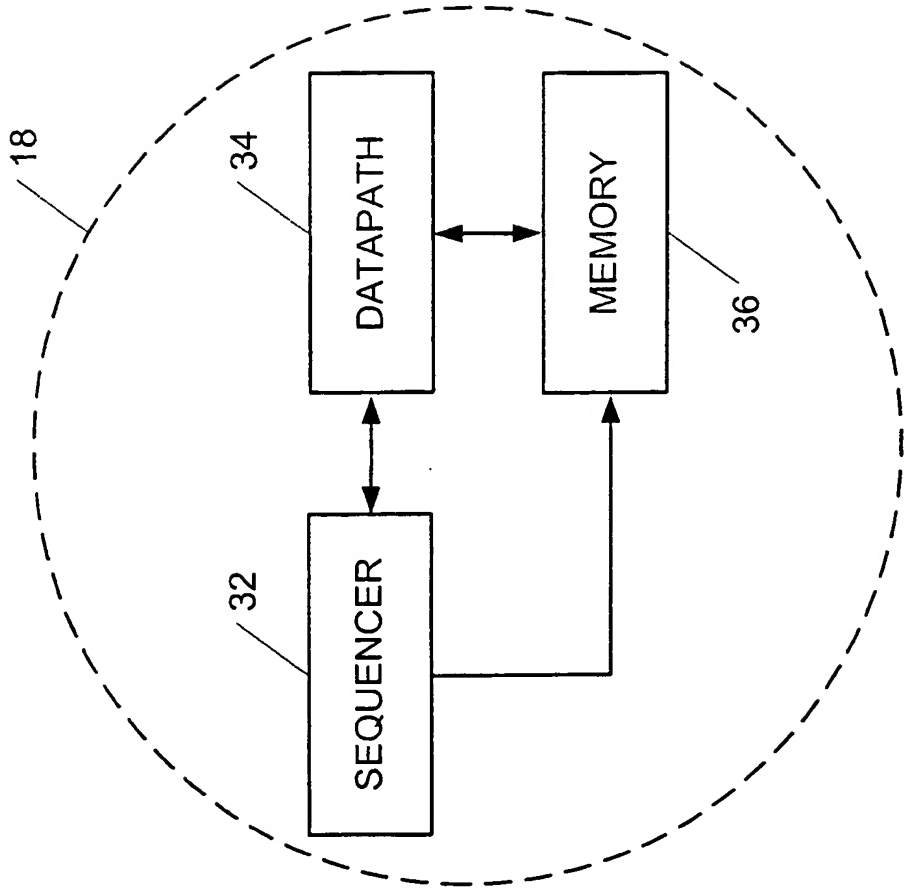

18
34
DATAPATH
MEMORY
36
32
SEQUENCER

FIG. 2B

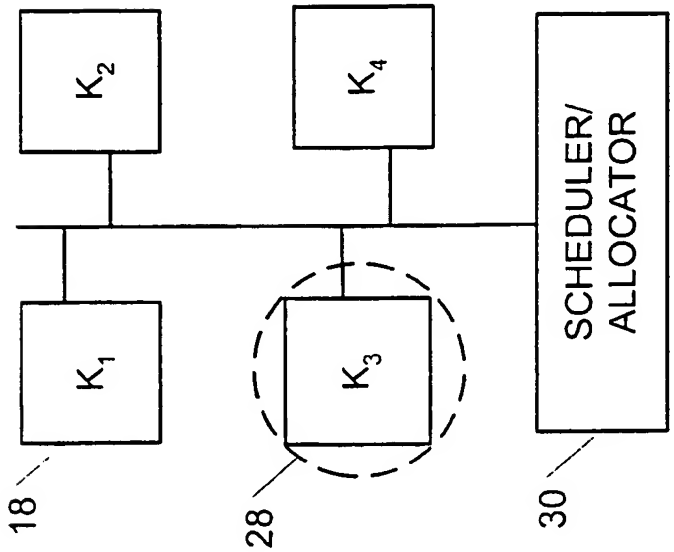

K2
K4
SCHEDULER/
ALLOCATOR
K1
K3
18
28
30

FIG. 2A

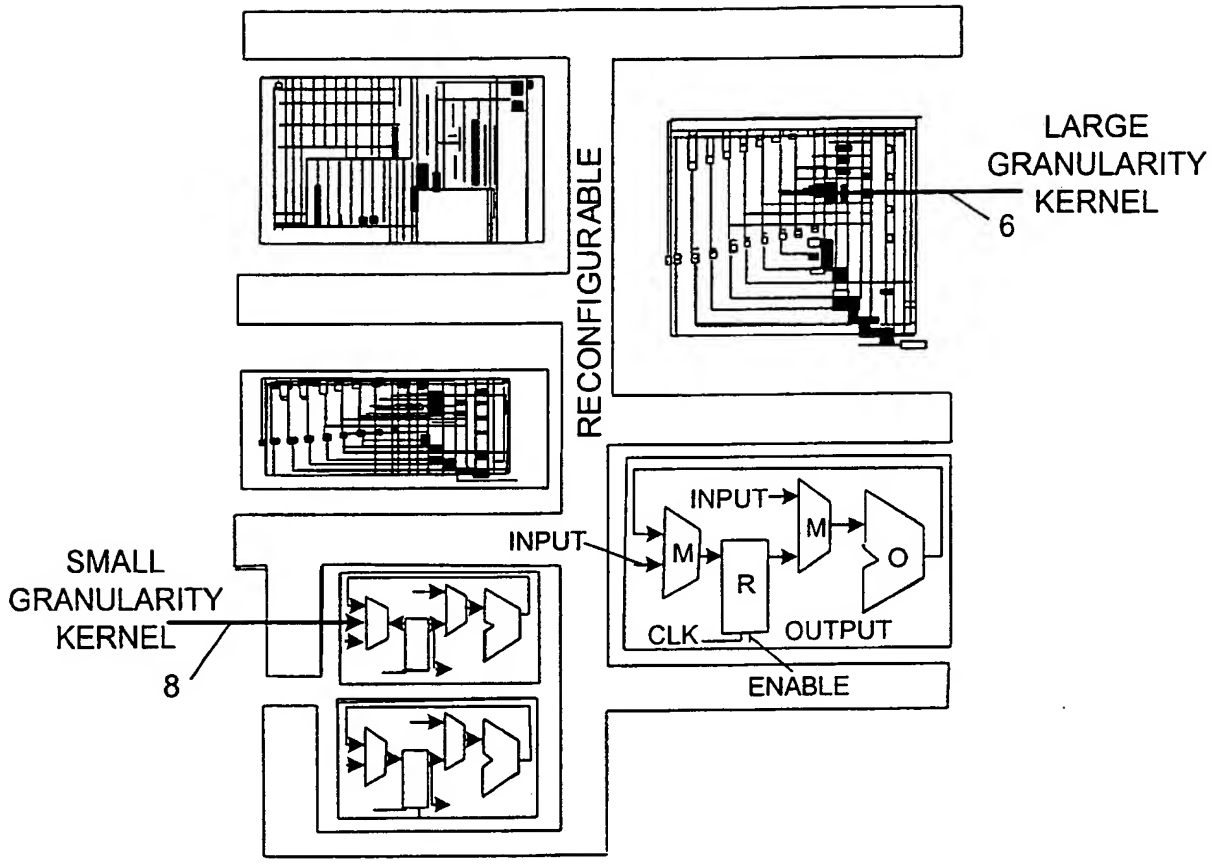

RECONFIGURABLE
LARGE
GRANULARITY
KERNEL
6
INPUT
M
M
R
O
CLK
OUTPUT
ENABLE
INPUT
SMALL
GRANULARITY
KERNEL
8

FIG.3

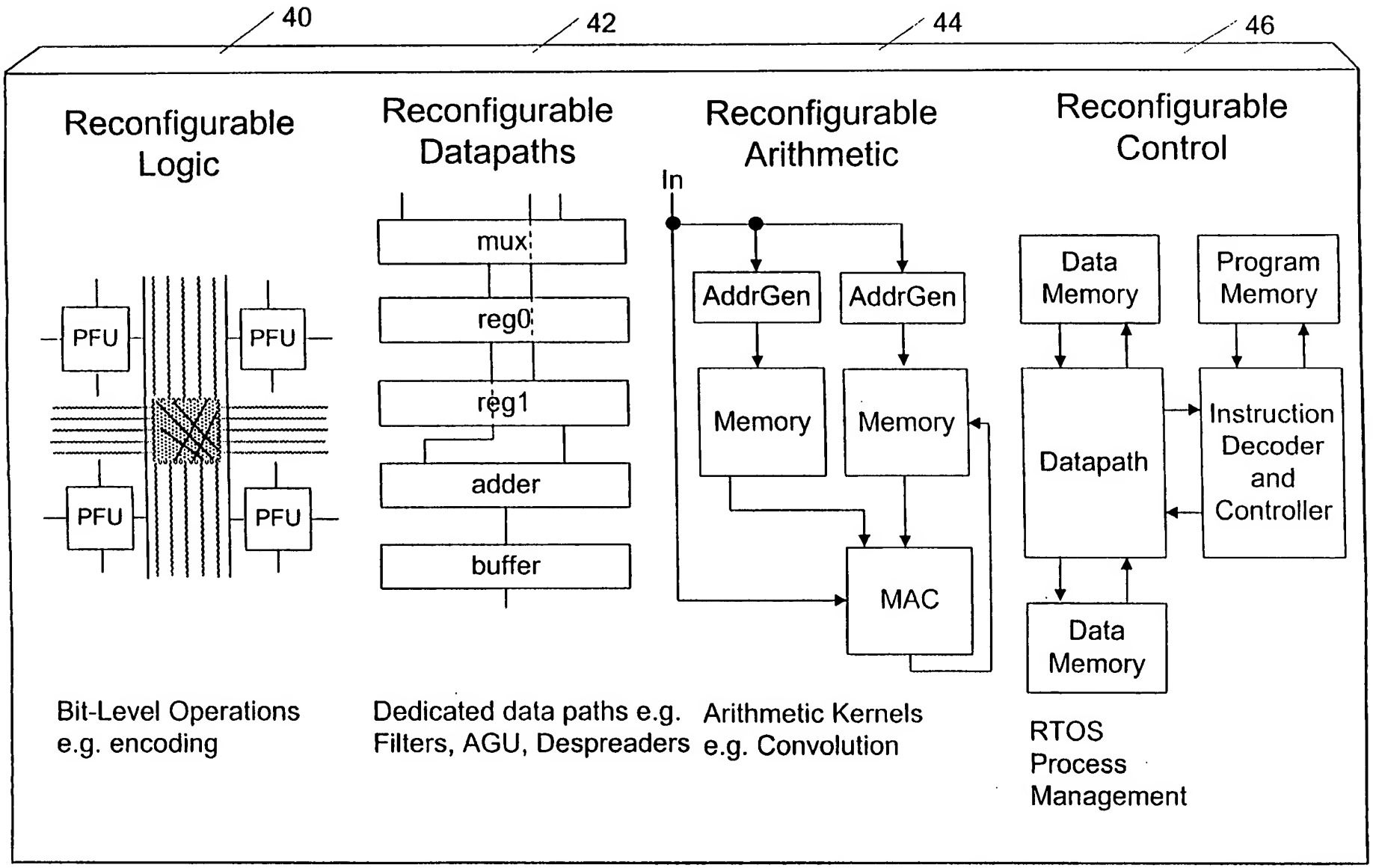


FIG. 4

PCT/US01/02982

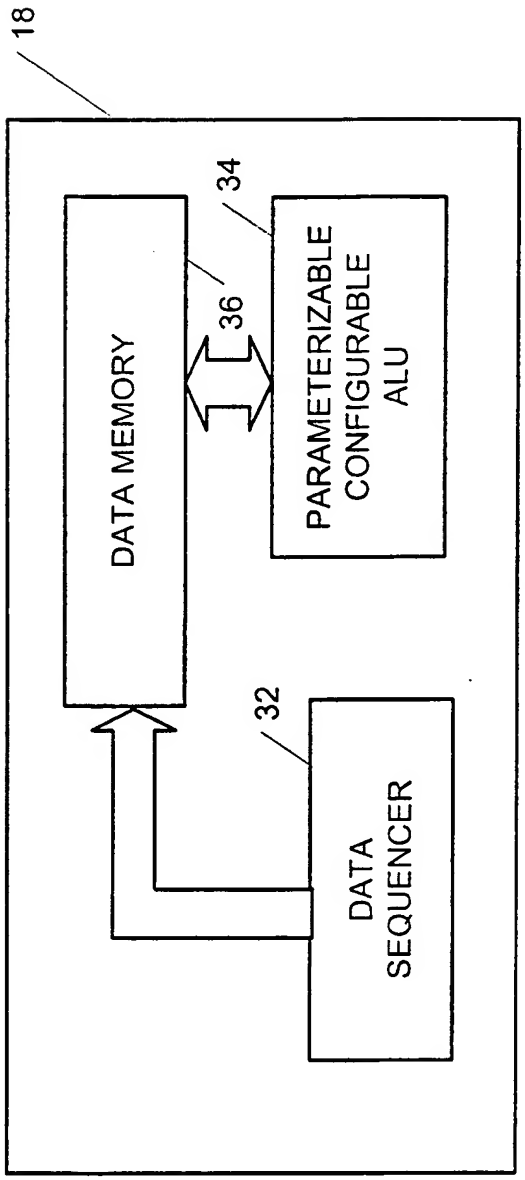

18
34
36
DATA MEMORY
PARAMETERIZABLE CONFIGURABLE ALU
32
DATA SEQUENCER

FIG. 5

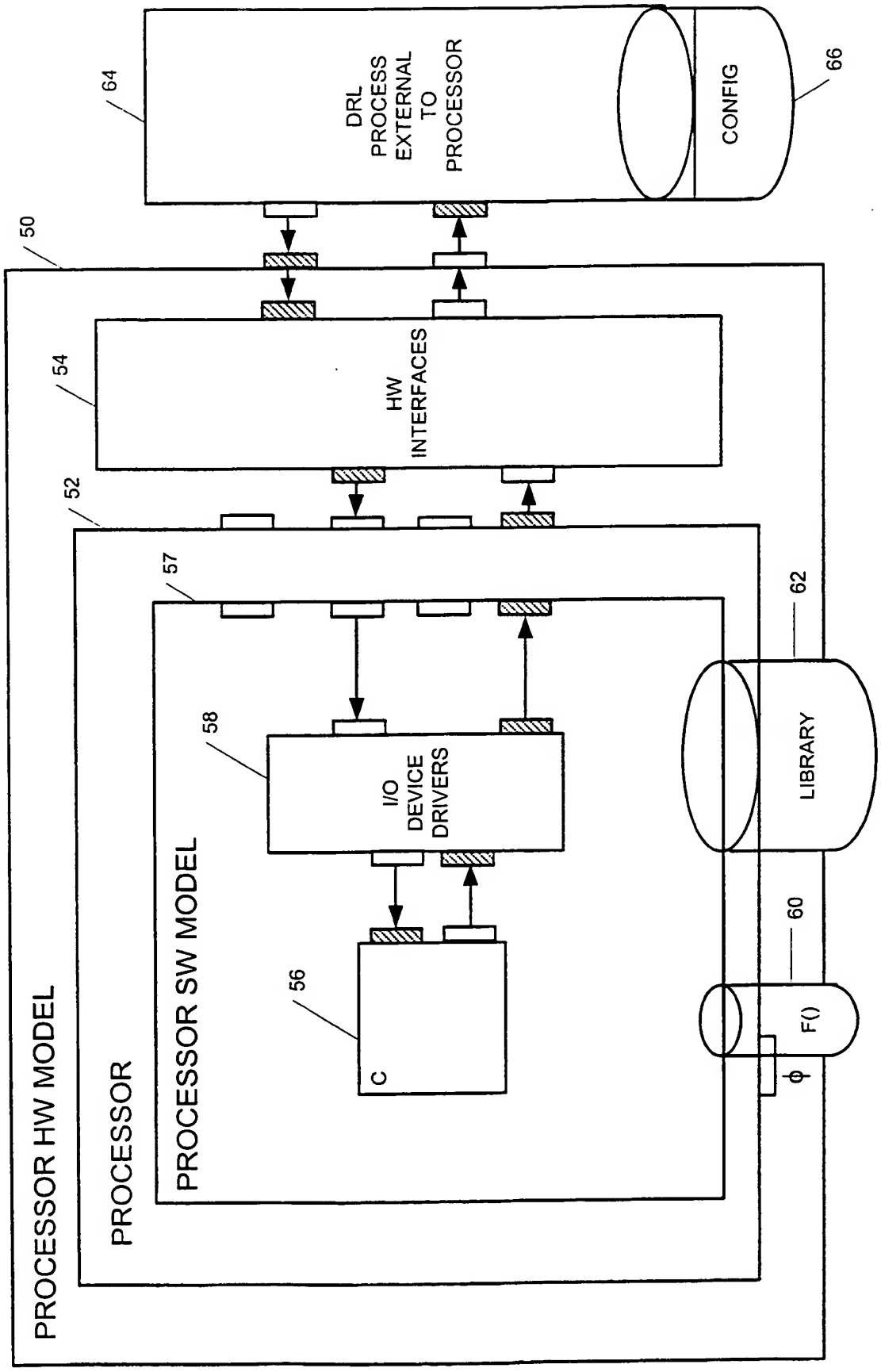

64
DRL PROCESS EXTERNAL TO PROCESSOR
CONFIG
66
50
54
HW INTERFACES
52
57
62
58
I/O DEVICE DRIVERS
LIBRARY
60
56
C
F()
ϕ
PROCESSOR HW MODEL
PROCESSOR
PROCESSOR SW MODEL

FIG.6

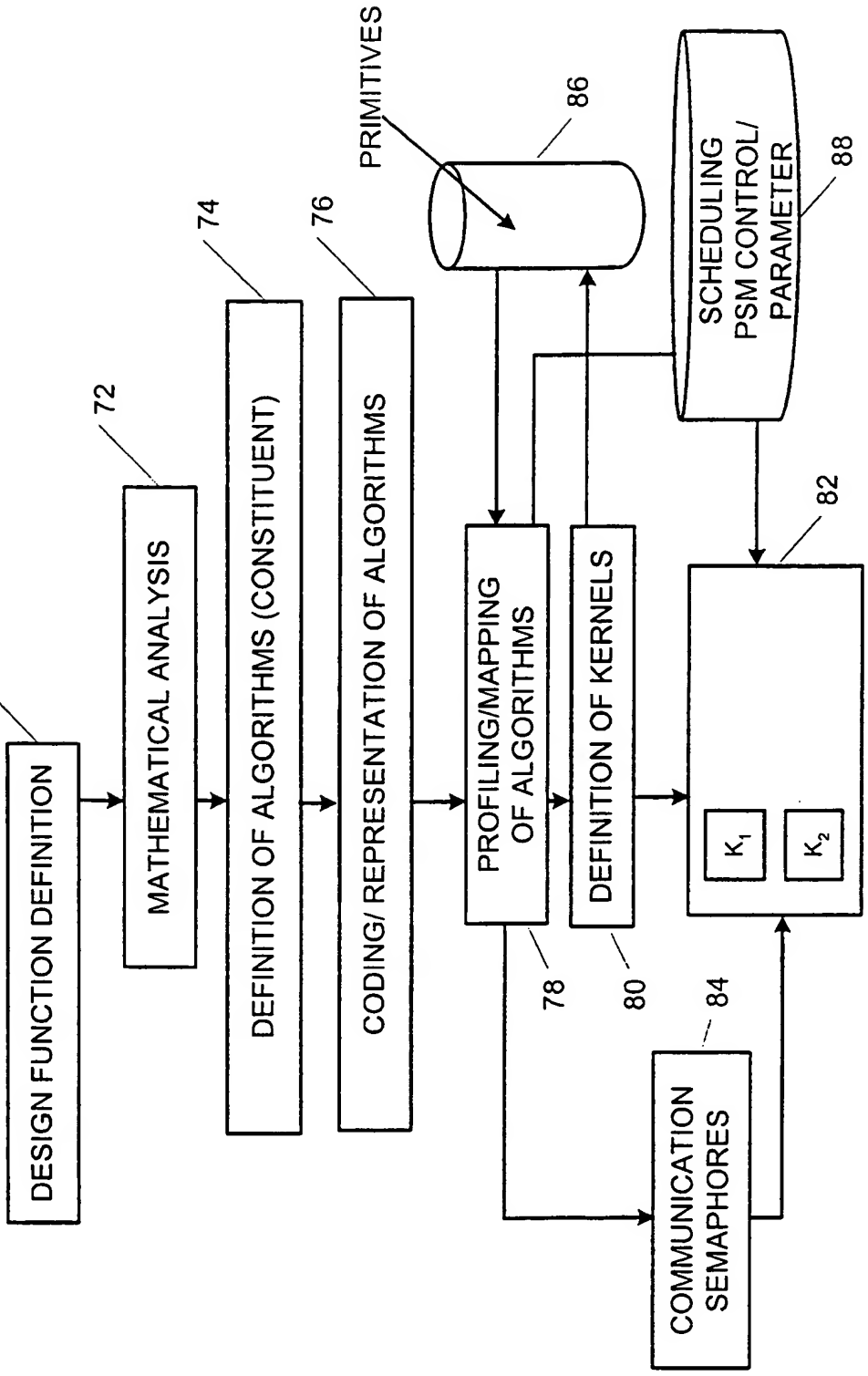

DESIGN FUNCTION DEFINITION
70
MATHEMATICAL ANALYSIS
72
DEFINITION OF ALGORITHMS (CONSTITUENT)
74
CODING/ REPRESENTATION OF ALGORITHMS
76
PRIMITIVES
86
SCHEDULING PSM CONTROL/ PARAMETER
88
PROFILING/MAPPING OF ALGORITHMS
78
DEFINITION OF KERNELS
80
82
K1
K2
COMMUNICATION SEMAPHORES
84

FIG. 7

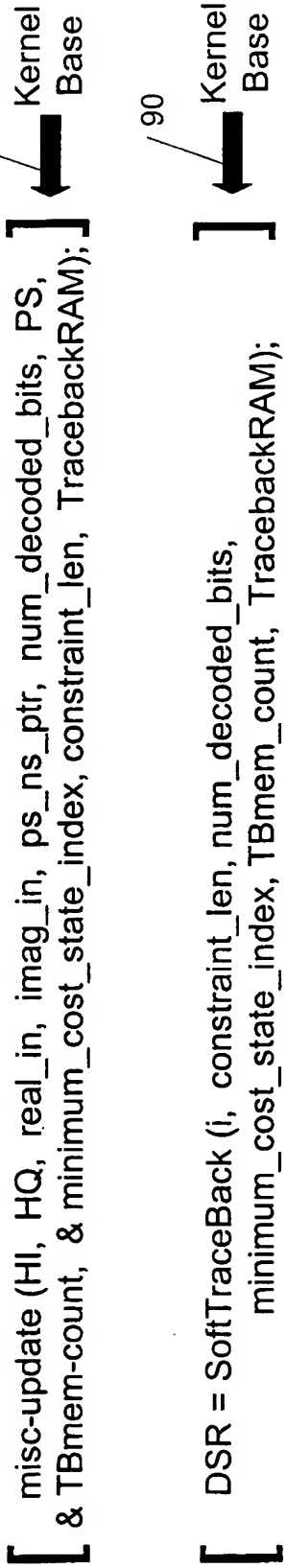

misc-update (HI, HQ, real_in, imag_in, ps_ns_ptr, num_decoded_bits, PS,
& TBmem-count, & minimum_cost_state_index, constraint_len, TracebackRAM);
90
Kernel Base
DSR = SoftTraceBack (i, constraint_len, num_decoded_bits,
minimum_cost_state_index, TBmem_count, TracebackRAM);
90
Kernel Base

FIG. 8

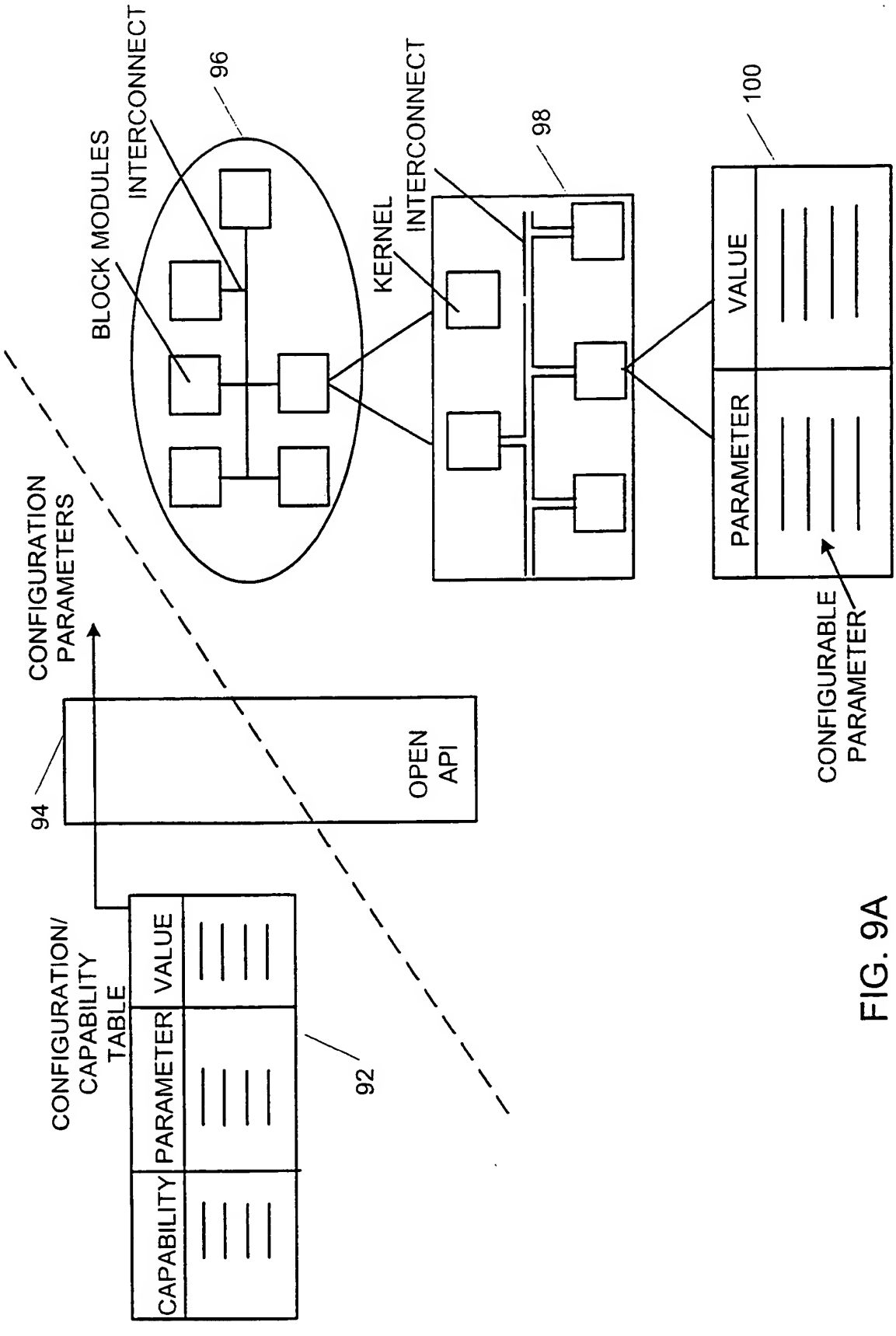

CONFIGURATION/
CAPABILITY
TABLE
CAPABILITY
PARAMETER
VALUE
92
94
OPEN
API
CONFIGURATION
PARAMETERS
BLOCK MODULES
INTERCONNECT
96
KERNEL
INTERCONNECT
98
100
PARAMETER
VALUE
CONFIGURABLE
PARAMETER

FIG. 9A

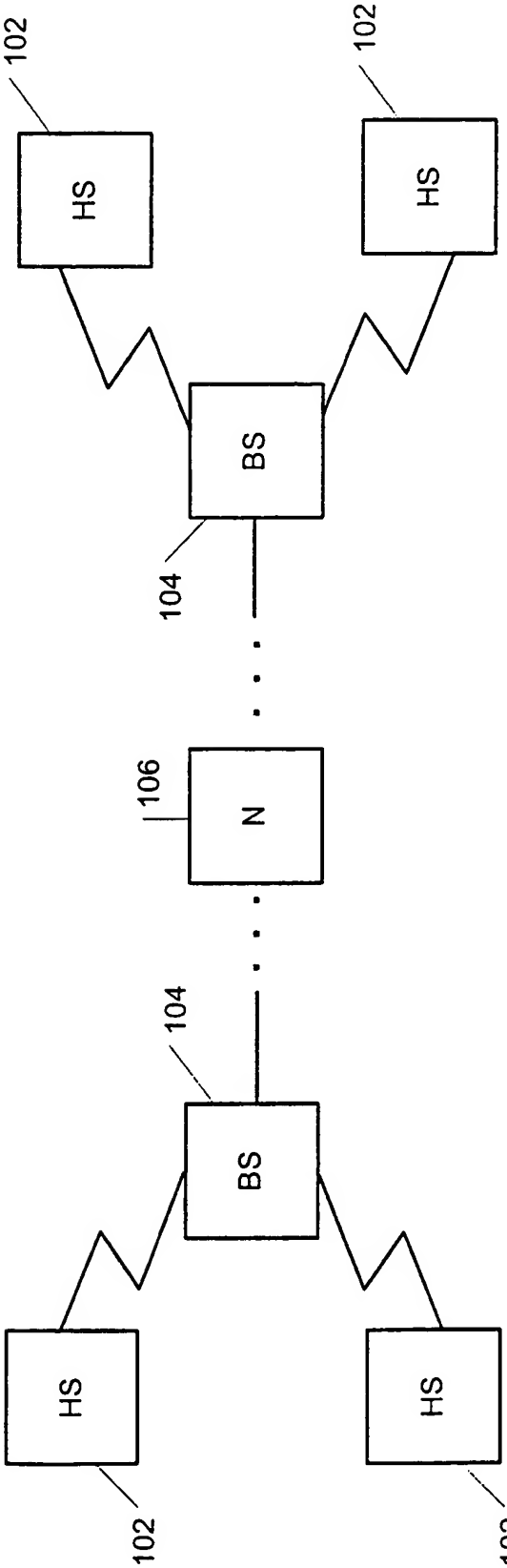


FIG. 9B

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US01/02982

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/50

US CL : 716/18, 1

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 716/18, 1, 17

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST, IEEE

trread same kernel same system design; thread same kernel same function$ same process$; multiple thread or mult?thread$2; cad; temporal constraint and kernel; temporal function and kernel; base station and kernel and thread

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 5,870,588 A (ROMPAEY et al.) 09 February 1999 (09.02.1999), column 1, lines 18-25; column 3, lines 5-67; column 4, lines 1-35; column 4, lines 57-63; column 6, lines 57-67; column 7, lines 1-31; column 11, lines 12-67; column 12, lines 1-34; column 13, lines 19-67; column 14, lines 1-40; column 16, lines 6-65; column 17, lines 25-67; column 26, lines 50-67; column 27, lines 1-3; Figs. 1, 2, 3, 10-18.	1-5, 7-9, 11-14 ---------- 6, 10, 15
Y	US 4,821,220 A (DUISBERG) 11 April 1989 (11.04.1989), see abstract; column 2, lines 56-68; column 3, lines 68; column 4, lines 1-11.	10, 15
Y	US 5,537,226 A (WOLBERG et al.) 16 July 1996 (16.07.1996), column 8 , lines 45-54; column 10, lines 6-23; column 13, lines 40-50.	6, 15
A	US 5,519,867 A (MOELLER et al.) 21 May 1996 (21.05.1996), entire document.	1-15
A, P	US 6,112,020 A (WRIGHT) 29 August 2000 (29.08.2000), entire document.	1-15
A	Bernard K. Gunther, "Multithreading with Distributed Functional Units," IEEE TRANSACTIONS ON COMPUTER, VOL. 46, NO. 4, April 1997, pages 399-411.	1-15

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

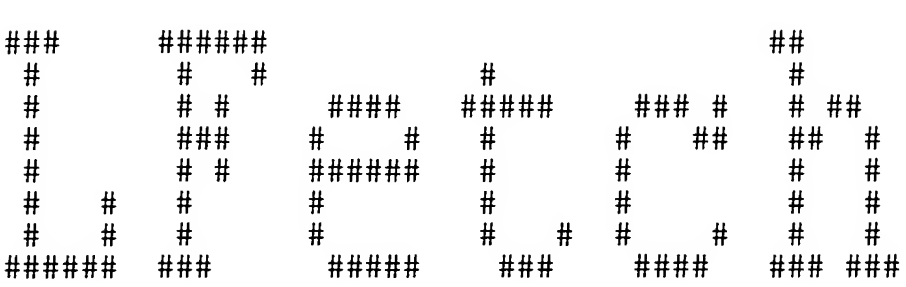
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

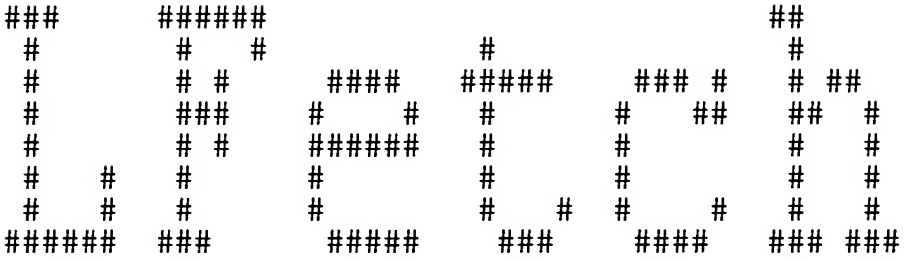
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
	11 APR 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Matthew Smith Telephone No. (703) 308-1782

Form PCT/ISA/210 (second sheet) (July 1998)

****
****





****
****

Print Job Information:

Date: 11/7/2006

Time: 3:33:45 PM

Job Number: 461